The Challenge of Multilinguality in Europeana: Web Services as Language Resources

Luca Dini[1] and Vivien Petras[2]

[1] *CELI, IT*
[2] *Humboldt University Berlin, DE*

The Challenge of Multilinguality in Europeana: Web Services as Language Resources

Luca Dini (CELI, IT) & Vivien Petras (Humboldt University Berlin, DE)

Abstract: Europeana has to face the tremendous challenge of providing multilingual functionalities for at least 10 languages (within the project phase of EuropeanaConnect, ultimately as many as official European languages (23)). It should be noted that these functionalities are particularly relevant for a multimedia collection such as Europeana: if accessing the full text of documents might raise obstacles due to language understanding, this is not the case for image or music material, which, however, is still only searchable in the language of the descriptions provided. The challenges that Europeana has to face concern several aspects of the lifecycle of language resources (the production phase being outside Europeana's scope) and can be grouped in the following way: (i) quality assessment and selection, (ii) integration, (iii) maintenance, (iv) licensing.

Challenges

Quality assessment and selection

In the first phase of the project an extensive scan of the available language resources was performed: these included both "free" resources and licensable resources available from project partners. These resources included both "low level" resources such as lexicons and morphological analyzers, and "high level" resources such as thesauri or bilingual dictionaries. Such a list immediately poses the problem of selection: Which resource is better suited to achieve Europeana goals? For some resource types, the criteria are set by a commonly agreed gold standard, e.g. POS tagging benchmarks. However, anybody who ever participated to an evaluation experiment is aware of how much time is spent on making in-house tools (let alone *third party resources*) compliant with competition standards. Even worse, for some crucial resources, such as *bilingual lexicons,* gold standards are just missing, and probably even difficult to conceive. The selection is therefore dependent on an application-based evaluation, which implies at least two steps:

- To set up an application dependent gold standard, trying to foresee application requirements: time-consuming but feasible.
- To integrate *each* resource for testing in the evaluation workflow: this is where the real problems start, as will emerge from the next section.

Integration

Facing the integration problem for such a number of languages immediately raises the problem of lack of *homogeneity/standards*:

For processing modules:

- Different capabilities;
- Different output formats;
- Different operating systems;
- Different programming languages

For static resources:

- Different linguistic assumptions;
- Different tag set;
- Different syntax;
- Different coverage

If these problems can be overcome in a the context of the integration of several resources with a single application, they become hardly tractable when integrating different resources *with each other*, which is often the case when dealing with individual open source resources or for the integration of general purpose and domain specific resources.

Maintenance

It is a well-known fact that maintaining a single resource is already an expensive task. In the case of Europeana, long-term maintenance could become a critical factor. Even at a "low" computational level, the presence of processing modules written in different programming languages and running on different operating systems is already a technical challenge. Maintenance becomes, however a *real* challenge when taking into account that a digital library is by default a changing environment. Even without pursuing the perfect "up-to-date-ness", resources need to be updated and enriched at regular intervals, especially as far as

terminology is concerned. Assuming the existence of a perfect lifecycle for resource maintenance, (which, for some type of resources, is not the case), this means, in the long run, the existence of a department of *at least* 23 mother-tongue linguists who are will maintain each resource in their language.

Licensing

The heterogeneity of licensing schemas also raises problems for integration. In the case of not-for-free resources negotiations are often difficult even for single resources, with a lot of contracting work behind. The case of "free" resources is, however, not much easier, as it not only requires to find a reasonable path across the constellation of different "free" licenses, but also to understand what the legal consequences of the integration of two resources with different licensing schemas are.

Can Web Services support Multilinguality in Europeana?

Work package 2 of the project EuropeanaConnect has the goal of providing multilingual acces to Europeana. In order to face the complex situation described above, a radically service-oriented view of language resources is proposed. The basic idea is that any access of the Europeana infrastructure to language information is mediated by a standardized web service: this includes both multilingual information "strictu sensu" (needed for indexing, searching, managing classification schemas, etc.) such as lemmatization, Named Entity extraction, thesauri look-up etc. and cross-lingual information (basically bilingual dictionary look-up). For any high level functionality there should be a public WSDL describing the interface that a certain web service should have in order to be compliant with Europeana. In this way we expect to trigger a dynamic and distributed involvement of both project partners and external parties.
While in architectural terms the advantages of such an approach are evident, the focus of this paper is to analyses in which sense they help to overcome part of the obstacles mentioned in the previous section.

Quality assessment and choice

From a technical point of view, the adoption of Web services does not change the complexity and fuzziness of the evaluation process. However, they allow a loosely coupled resource wrapping, a less invasive and more agile process than local resource wrapping or, even worse, resource transformation. Moreover, in a competitive setting, they simplify the process of benchmarking: anyone proposing a better solution for Europeana is enabled to prove its claim just by implementing the relevant Web service, which could then be directly connected to the evaluation workflow, without any additional overhead.

Integration

Concerning the integration of processing modules, it is clear that the Web service paradigm will solve most of the problems raised above, in particular:

- *Different capabilities*. Web services are able to declaratively encode the capabilities they have. The consuming application can therefore be tailored on that.
- *Different output format*s. Translating a proprietary/unusual format into a public specification and exposing it to the external word (under whatever licensing schema) is a much more convenient operation than producing a wrapper for the sake of one application.
- *Different operating systems or programming languages*: these two barriers to integration are overcome by definition, as the service can stay on the most adequate platform without enforcing any requirement to the calling application.

More interesting is the case of "static" resources integrated into a Web service, thus accessible as if they were a processing module. In this case the biggest advantage brought by the paradigm is in terms of cleanness and reversibility of operations. The resource doesn't need to be integrated, but stays as it is. The wrapping web service just takes care of mapping the resource in the desired Web service format. The advantages of this process are:

- The resource is not transformed in any sense, which allows seamless integration of successive versions and minimizes conversion errors.
- Any upgrade of the resource is immediately reflected by the quality of the service.

- Native (i.e. third party delivered) wrappers can be directly used without jeopardizing the non- functional features of the Europeana central system.
- There is the possibility that an Europeana compliant service is produced directly by third parties willing to participate to the common effort.
- Mapping among different tag sets can be realized as a public harmonization service.

These advantages do not constitute an answer to the problem of *integration of different resources*: for that we probably have to look towards *web service composition*. This implies a shift from the view of language resources integration as a *merging process* to the one of a *business flow*. Under this view the process of language integration is "reduced" to the identification of possible preconditions of flow and access priorities to different functionally equivalent resources. The interesting thing is that these processes can be modelled in a declarative way thanks to the availability of several Web Service Composition Standards such as BPEL4WS, BPML, WSCI, DAML-S etc.: changing the integration flow (e.g. for testing different combinations or answering different functionalities) becomes consequently a matter of modifying an XML file describing the business logic.

Maintenance

Web services cannot provide a solution to the maintenance problem on a technical ground but they can ease the process of *outsourcing*. As a service is by definition maintained by a service provider, Europeana could benefit from the opportunity of selecting which resources can be maintained internally and which ones can be just assigned as external web services, which might reduce maintenance costs and increase the quality.

Licensing

One of the typical weak aspects in trading language resources is the lack (or the fuzziness) of licensing schemas. This is partly due to language resources being "strange" objects with a niche market and with poor competition rate, which inhibits the consolidation of acknowledged licensing schema. With the vision of "language resources as Web services" some light could be shed on the licensing issues, simply by importing practices from the business web service community, especially in the area of multimedia, business information and geographic information.

Conclusions

Besides the obvious conclusion that web services could prove to be beneficial for initiatives of the linguistic complexity of Europeana, there is an aspect, which needs to be made more explicit: the desperate need for standards. Europeana could invent its own standards for communication among language resources or adapt some of the already available "protocols" (such as UIMA, which offers only a syntactic layer) but this situation should rather be discouraged, for an evident reason: Web services need an *environment* and a *community* and in order to be effective need to be re-used: a Web service with only one client (however big) makes little sense. It would be a waste of time and money if the initiatives currently under discussion and execution could not find a way of harmonizing data format and protocols.